Lightning talk: Query your Cloud Infrastructure interactively with Steampipe!

ITLT session #24 - 2023-06-09: https://indico.cern.ch/event/1280138/

Well-known tool: jq

Allows filtering, modifying and even creating JSON on the command line

Examples:

```
$ openstack server list -f json
[
  {
    "ID": "06f3a71e-ecc5-42da-94f4-2af8646036a5",
    "Name": "webeos-proto-master-tzv8k",
    "Status": "ACTIVE",
    "Networks": {
      "CERN_NETWORK": [
        "188.185.101.143",
        "2001:1458:d00:4c::100:191"
      ]
    },
    "Image": "fedora-coreos-33.20210217.3.0-openstack.x86_64",
    "Flavor": "m2.xlarge"
  },
  {
    "ID": "90d7b567-e95c-40f9-be12-68d52d8438af",
    "Name": "infra-4rvnj",
    "Status": "ACTIVE",
    "Networks": {
      "CERN_NETWORK": [
        "137.138.151.106",
        "2001:1458:d00:12::3bb"
      ]
    },
    "Image": "fedora-coreos-33.20210217.3.0-openstack.x86_64",
    "Flavor": "m2.xlarge"
  },
...
```

```
# show all flavors in use and count occurences
openstack server list -f json | jq '.[] | .Image' | sort | uniq -c
      4 ""
      4 "fedora-coreos-33.20210217.3.0-openstack.x86_64"
      7 "fedora-coreos-34.20210904.3.0-openstack.x86_64"
      9 "fedora-coreos-35.20220327.3.0-openstack.x86_64"
     60 "fedora-coreos-36.20220716.3.1-openstack.x86_64"
      6 "fedora-coreos-37.20221127.3.0-openstack.x86_64"
```

```
# get all containers that do not use an image from the Harbor registry
$ oc get pods -A -o json | \
    jq -r '.items[].spec.containers[] | select(.image | startswith("registry.cern.ch") | not) | "\(.name): \(.image)"'
| \
    sort -u
authz-operator: gitlab-registry.cern.ch/paas-tools/operators/authz-operator:RELEASE.2023.03.03T14-25-39Z
backups-volumes-cephfs: gitlab-registry.cern.ch/paas-tools/okd4-deployment/backup-cephfs-volumes:RELEASE.2023.04.04T12-
36-10Z
cephfs-provisioner: k8s.gcr.io/sig-storage/csi-provisioner:v3.0.0
cephfs-reclaim-deleted-volumes: gitlab-registry.cern.ch/paas-tools/okd4-deployment/reclaim-cephfs-
volumes:RELEASE.2021.02.02T17-07-38Z
cephfs-registrar: k8s.gcr.io/sig-storage/csi-node-driver-registrar:v2.4.0
cephfs-resizer: k8s.gcr.io/sig-storage/csi-resizer:v1.3.0
cephfs-snapshotter: k8s.gcr.io/sig-storage/csi-snapshotter:v5.0.1
cert-manager: quay.io/jetstack/cert-manager-cainjector:v1.5.5
cert-manager: quay.io/jetstack/cert-manager-controller:v1.5.5
cert-manager: quay.io/jetstack/cert-manager-webhook:v1.5.5
csi-provisioner: registry.k8s.io/sig-storage/csi-provisioner:v3.2.1
csi-resizer: registry.k8s.io/sig-storage/csi-resizer:v1.5.0
csi-snapshotter: registry.k8s.io/sig-storage/csi-snapshotter:v6.0.1
dns-synchronizer: gitlab-registry.cern.ch/paas-tools/okd4-deployment/dns-hostname-synchronizer:RELEASE.2023.04.17T09-
58-28Z
driver-registrar: registry.k8s.io/sig-storage/csi-node-driver-registrar:v2.5.1
eos-volume-provisioner: gitlab-registry.cern.ch/paas-tools/okd4-deployment/eos-volume-
provisioner:RELEASE.2021.05.07T15-22-27Z
external-dns: registry.k8s.io/external-dns/external-dns:v0.13.2
fluentd-aggregator: gitlab-registry.cern.ch/paas-tools/okd4-deployment/log-forwarding:RELEASE.2020.11.25T14-31-56Z
garbage-collector: image-registry.openshift-image-registry.svc:5000/openshift/cli:latest
landb-operator: gitlab-registry.cern.ch/paas-tools/operators/landb-operator:RELEASE.2022.11.30T16-45-30Z
node-problem-detector: k8s.gcr.io/node-problem-detector/node-problem-detector:v0.8.7
openshift-sls-availability: gitlab-registry.cern.ch/paas-tools/okd4-deployment/openshift-sls-
availability:RELEASE.2022.08.03T13-39-24Z
provisioner: k8s.gcr.io/sig-storage/csi-provisioner:v3.2.1
publish-dns: gitlab-registry.cern.ch/paas-tools/okd4-deployment/ingress-publish-dns:RELEASE.2023.04.04T12-34-04Z
registrar: registry.k8s.io/sig-storage/csi-node-driver-registrar:v2.5.1
registry-server: registry.access.redhat.com/redhat/community-operator-index:v4.11
sitedetails-operator: gitlab-registry.cern.ch/paas-tools/operators/sitedetails-operator:RELEASE.2023.05.08T16-05-52Z
user-alerts-adapter: gitlab-registry.cern.ch/paas-tools/okd4-deployment/user-alerts-adapter:RELEASE.2023.03.08T10-57-
46Z
velero: velero/velero:v1.8.1
webservices-blocker-controller: gitlab-registry.cern.ch/paas-tools/okd4-deployment/blocker-
controller:RELEASE.2023.01.25T12-40-44Z
```

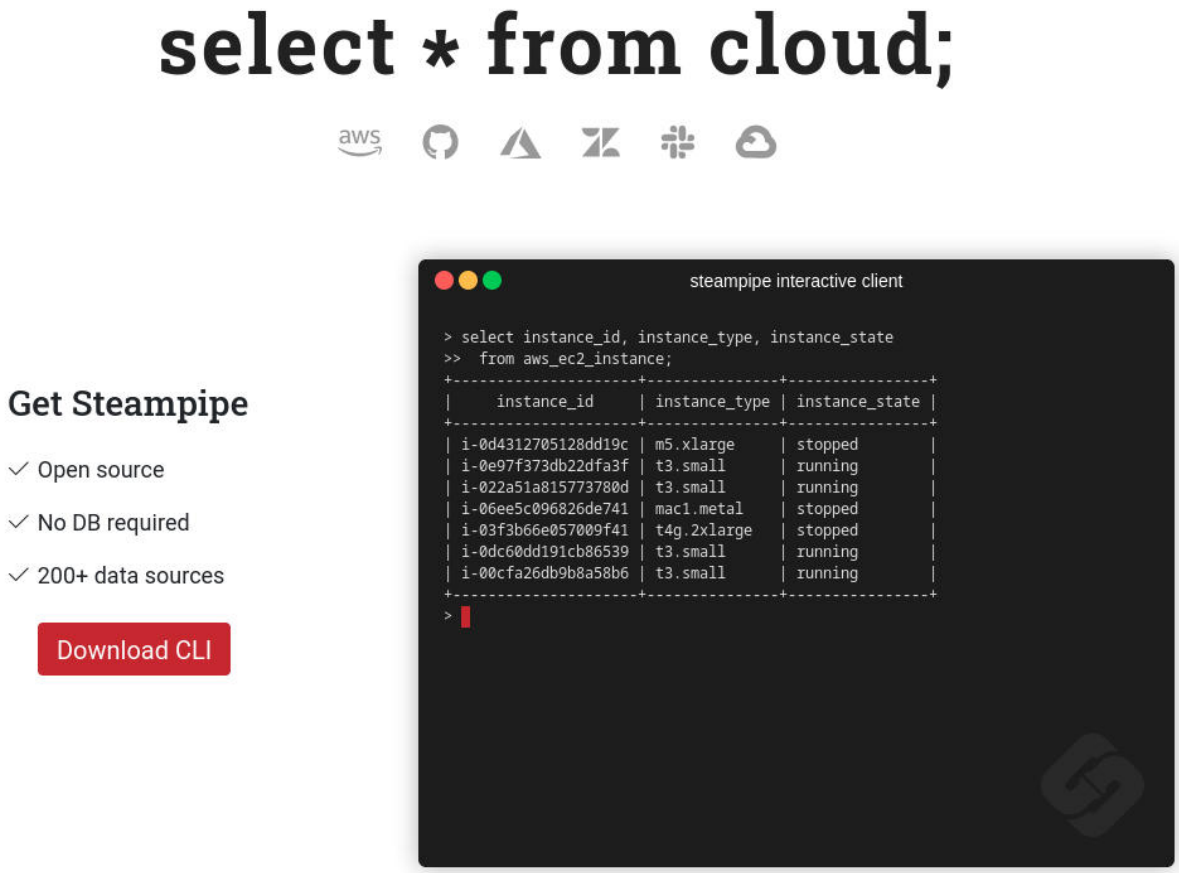
Another recent real-world example:

```
# get full JSON of all PVCs with type "cvmfs"
oc get pvc -A -o json | jq -r '{items: [.items[] | select(.spec.storageClassName | startswith("cvmfs-")?) | .]}' >
cvmfs-pvcs.json

# get all running pods
oc get pods -A --field-selector=status.phase=Running -o json > running-pods.json

# obtain comma-separated and quoted list of all PVCs using CVMFS
cvmfs_pvcs_list=$(jq -c '[.items[]| .metadata.name]' cvmfs-pvcs.json | tr -d \[\])

# identify on which nodes these volumes are mounted
nodes_with_cvmfs=$(jq -r '.items[] | select(.spec.volumes[].persistentVolumeClaim.claimName | IN($cvmfs_pvcs_list)) |
.spec.nodeName' running-pods.json | sort -u)
```

We can see that `jq` is really handy for performing quick look ups, there are **two drawbacks**:

1. data always flows linearly from left to right, and all data needs to be present in the initial input (no dynamic lookups)
2. the query language is completely custom and needs to learned from scratch (usually including a lot of web searches…)

Introducing: Steampipe



Allows querying “Cloud” resources with regular SQL

Relevant data sources :

- Cloud providers: AWS, Azure, DigitalOcean, Equinex, Fly.io, GCP, Hetzner, IBM Cloud, Linode, OVH
- **Kubernetes**, **Keycloak**, **LDAP**, Nomad
- Local files: INI, JSON, YAML, CSV
- VCS: **GitLab**, GitHub, Bitbucket
- Jira, Confluence
- Microsoft 365

Set up

```
# https://steampipe.io/downloads
$ yay -S steampipe

$ steampipe -v
Steampipe v0.20.3

$ steampipe plugin install turbot/kubernetes theapsgroup/gitlab
$ export KUBE_CONFIG_PATHS=
$ export GITLAB_ADDR=https://gitlab.cern.ch/api/v4
$ export GITLAB_TOKEN=<YOUR_GITLAB_PAT>
```

Let's give it a try:

```
# command mode:
$ steampipe query "select * from gitlab_version;"
+------------+-------------+-----------------------------+
| version    | revision    | _ctx                        |
+------------+-------------+-----------------------------+
| 15.10.8-ee | 9ec5f337f9f | {"connection_name":"gitlab"} |
+------------+-------------+-----------------------------+

# interactive mode
$ steampipe query
> select * from gitlab_version;
+------------+-------------+-----------------------------+
| version    | revision    | _ctx                        |
+------------+-------------+-----------------------------+
| 15.10.8-ee | 9ec5f337f9f | {"connection_name":"gitlab"} |
+------------+-------------+-----------------------------+
```

Show all available tables and describe them:

```
> .tables
 ==> gitlab
+-----------------------------------+-------------------------------------------------------------------------------
-----------------------------------+
| table                             | description
|
+-----------------------------------+-------------------------------------------------------------------------------
-----------------------------------+
| gitlab_application                | Obtain information about OAuth applications within the GitLab instance.
|
| gitlab_branch                     | Obtain information on branches for a specific project within the GitLab instance.
|
| gitlab_commit                     | Obtain information about commits for a specific project within the GitLab
instance.                                 |
| gitlab_epic                       | Obtain information about epics for a specific group within the GitLab instance.
|
| gitlab_group                      | Obtain information about groups within the GitLab instance.
|
...

> .inspect gitlab_issue
+-----------------------+--------------------------+-------------------------------------------------------------------
---------------+
| column                | type                     | description
|
+-----------------------+--------------------------+-------------------------------------------------------------------
---------------+
| assignee              | text                     | The username of the user assigned to the issue - link to
`gitlab_user.username`  |
| assignee_id           | bigint                   | The ID of the user assigned to the issue - link to
`gitlab_user.id`.             |
| assignees             | jsonb                    | An array of assigned usernames, for when more than one user is
assigned.         |
```

Examples

List all issues that are assigned to me:

```
> select title,labels from gitlab_issue where assignee = 'jhensche' and state = 'opened';
+-------------------------------------------------------------------------+-----------------------------------------------
-----------------------+
| title                                                                   | labels
|
+-------------------------------------------------------------------------+-----------------------------------------------
-----------------------+
| Document web redirector operations and troubleshooting                  | ["P::3"]
|
| WR v2 CI improvements                                                   | []
|
| Document steps for high-availability in PaaS user docs                  | ["Area::docs","P::4","Project::PaaS","To Do"]
|
| Improve secrets management: switch from Gitlab CI variables to Vault | ["Area::OKD4 deployment","In
progress","P::2","Project::OKD4 infra"] |
+-------------------------------------------------------------------------+-----------------------------------------------
-----------------------+
```

Hmm, I would also like to see the project name, but the table only has a `project_id` column (which is not very helpful).

Let's resolve the ID to a name with SQL!

```
> SELECT title,labels,gitlab_my_project.name
FROM gitlab_issue
LEFT JOIN gitlab_my_project
ON project_id = gitlab_my_project.id
WHERE assignee = 'jhensche' AND state = 'opened';
+-------------------------------------------------------------------------+-----------------------------------------
----------------------+-----------------------+
| title                                                                   | labels
| name                   |
+-------------------------------------------------------------------------+-----------------------------------------
----------------------+-----------------------+
| WR v2 CI improvements                                                   | []
| jhensche               |
| Document web redirector operations and troubleshooting                  | ["P::3"]
| webframeworks-planning |
| Document steps for high-availability in PaaS user docs                  | ["Area::docs","P::4","Project::PaaS","To Do"]
| webframeworks-planning |
| Improve secrets management: switch from Gitlab CI variables to Vault | ["Area::OKD4 deployment","In
progress","P::2","Project::OKD4 infra"] | webframeworks-planning |
+-------------------------------------------------------------------------+-----------------------------------------
----------------------+-----------------------+
```

Show all issues that have been created in a project in the last week:

```
SELECT title,author,created_at FROM gitlab_issue WHERE project_id = 85447 AND created_at >= '2023-06-05';
+-------------------------------------------------+----------+---------------------------+
| title                                           | author   | created_at                |
+-------------------------------------------------+----------+---------------------------+
| Test webeos in Alma9                            | dchatzic | 2023-06-07T12:18:59+02:00 |
| Expore options for cleaning up test sites/previews | dchatzic | 2023-06-07T12:11:53+02:00 |
| webeos-site-operator: Add server version in UPD    | dchatzic | 2023-06-07T12:07:14+02:00 |
| Find alternative of php-fpm for .htaccess          | dchatzic | 2023-06-07T12:01:48+02:00 |
| Review Apache configuration                        | dchatzic | 2023-06-07T11:53:30+02:00 |
| Remove legacy-import from web-redirector-v2        | jhensche | 2023-06-06T09:38:31+02:00 |
| Prepare deployment for webeos in Alma9             | dchatzic | 2023-06-07T11:46:17+02:00 |
+-------------------------------------------------+----------+---------------------------+
```

Kubernetes examples

Show all Statefulsets in a cluster:

```
> select name,namespace from kubernetes_stateful_set;
+-------------------------------------------------+-----------------------------------+
| name                                            | namespace                         |
+-------------------------------------------------+-----------------------------------+
| prometheus-user-workload                        | openshift-user-workload-monitoring |
| fluentd-aggregator                              | openshift-logging                 |
| alertmanager-main                               | openshift-monitoring              |
| manila-csi-openstack-manila-csi-controllerplugin | openshift-cern-cephfs             |
| sonarqube-sonarqube                             | test-alex-demo-category           |
| prometheus-k8s                                  | openshift-monitoring              |
| thanos-ruler-user-workload                      | openshift-user-workload-monitoring |
| argocd-application-controller                   | openshift-cern-argocd             |
+-------------------------------------------------+-----------------------------------+
```

Let's find all pods that use a CVMFS volume:

```
> .inspect kubernetes_persistent_volume_claim
> .inspect kubernetes_pod
> SELECT name,namespace FROM kubernetes_persistent_volume_claim WHERE storage_class LIKE 'cvmfs%';
+--------------+---------------------+
| name         | namespace           |
+--------------+---------------------+
| cvmfs-bril   | brilview-diamantis  |
| cvmfs-cms-ib | test-alex-paas-stg2 |
| cvmfs-bril   | test-brilview       |
+--------------+---------------------+
```

Version 1:

```
> SELECT name,namespace FROM kubernetes_pod WHERE volumes #>> '{}' LIKE '%cvmfs%';
```

Version 2:

```
> SELECT name,namespace
FROM kubernetes_pod
WHERE EXISTS (
          SELECT TRUE
          FROM jsonb_array_elements(volumes) x
          WHERE x->'persistentVolumeClaim'->>'claimName' LIKE 'cvmfs%'
);
```

Version 3:

```
> SELECT name,namespace
FROM kubernetes_pod
WHERE EXISTS (
          SELECT TRUE
          FROM jsonb_array_elements(volumes) x
          WHERE x->'persistentVolumeClaim'->>'claimName' IN (
                    SELECT name
                    FROM kubernetes_persistent_volume_claim
                    WHERE storage_class LIKE 'cvmfs%'
          )
);
```

Leverage the full power of PostgreSQL: store the output in a temporary table!

```
> CREATE TEMPORARY TABLE namespaces_with_cvmfs AS (SELECT name,namespace,volumes
FROM kubernetes_pod
WHERE EXISTS (
          SELECT TRUE
          FROM jsonb_array_elements(volumes) x
          WHERE x->'persistentVolumeClaim'->>'claimName' IN (
                    SELECT name
                    FROM kubernetes_persistent_volume_claim
                    WHERE storage_class LIKE 'cvmfs%'
          )
));
SELECT * FROM namespaces_with_cvmfs;
```

Find out who the owner is:

```
> SELECT
    labels->>'lifecycle.webservices.cern.ch/owner' AS owner,
    labels->>'lifecycle.webservices.cern.ch/resourceCategory' AS category
  FROM kubernetes_namespace
  WHERE name IN (SELECT namespace FROM namespaces_with_cvmfs);
+----------+----------+
| owner    | category |
+----------+----------+
| alossent | Official |
| xiezhen  | Test     |
+----------+----------+
```